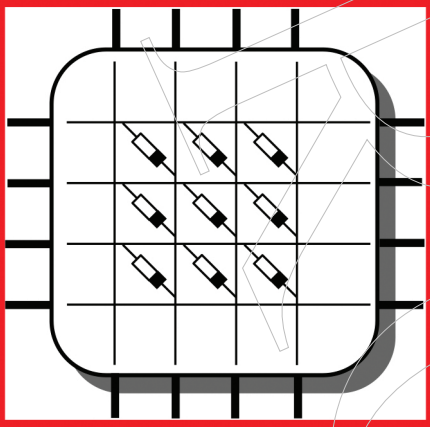
John Reuben



Resistive RAM & Peripheral Circuitry

An Integrated Circuit Perspective

John Reuben

Resistive RAM and Peripheral Circuitry

An Integrated Circuit Perspective

Bibliographic Information:

Book title:
Resistive RAM and Peripheral Circuitry
An Integrated Circuit Perspective

Author:
John Reuben, Ph.D
Chair of Computer Architecture
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Erlangen, Germany
ORCID: 0000-0002-7891-4975
Scopus Author ID: 55270477700

ISBN: 978-3-00-077848-3

Edition Number:
First Edition, Published February 2024

Number of Pages: 148
54 Color Illustrations

©2024 John Reuben

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. The rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical medium, now known or hereafter developed are held by the author.

NO AI TRAINING: Without in any way limiting the author's exclusive rights under copyright, any use of this publication to "train" generative artificial intelligence (AI) technologies to generate text is expressly prohibited. The author reserves all rights to license uses of this work for generative AI training and development of machine learning language models.

Bibliografische Information
(in German):

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über dnb.dnb.de abrufbar.

Die automatisierte Analyse des Werkes, um daraus Informationen insbesondere über Muster, Trends und Korrelationen gemäß §44b UrhG ("Text und Data Minin") zu gewinnen, ist untersagt.

ISBN: 978-3-00-077848-3

Preface

This book grew out of my seven years of post-doctoral research (2017-2023). In the year 2017, I held the Viterbi fellowship in Technion, Haifa, Israel. It was in Technion, I was first introduced to memristors. In those days, ReRAM (a type of memristor) was not manufactured and ReRAM was more a prototypical device, researched in a few labs in the world. From 2018 onwards, I was hosted at the Chair of Computer Architecture, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany. At FAU, Germany, I involved in various DFG-funded (German Research Foundation) projects focusing on ReRAM and FTJ (Ferroelectric tunnel Junctions). For ReRAM technology, we used the ReRAM manufactured at Institute for High Performance Microelectronics, Frankfurt Oder, Germany. Hence the book grew out of practical research conducted at some of the leading research labs and universities of the world.

This book is written as an introductory textbook* on ReRAM. Although other books on emerging Non-volatile memories (NVM) are available, there was a need for a specific book on ReRAM, given the market it has. Many existing books (to two of which I myself have contributed chapters) are a compilation of chapters from different research groups. Although comprehensive in coverage, such books lacked a cohesive flow. Moreover, they were written at a higher level since they were summarizing research in this field and were not written as a textbook for a university student. On the contrary, this book is written in an easy-to-understand tutorial manner. The purpose is to clarify basic concepts and introduce the reader to this field of emerging NVMs and ReRAM technology in particular. Hence, this book is written for university students considering a career in semiconductor industry. Although written with university students in mind, practicing engineers and researchers entering this field will find this book

* The contents of this book were prepared as course material for "Resistive RAM and In-Memory Computing" course offered at Chair of Computer Architecture, FAU, Erlangen.

immensely helpful because the book condenses the findings of decade-long research in this field into few chapters (≈ 150 pages). A preliminary search into any research database will show that this area of emerging NVM is very dynamic, with approximately 1000 papers published every year. As of 2020, the number of memristor-related articles published in technical literature is around 11000*. Referring to the most reliable technical literature (IEEE Transactions and other top journals), I have not only presented the recent research in a concise manner, but also simplified it for easy reading.

Almost all major semiconductor companies are investing heavily in non-volatile memory design and research. Furthermore, due to the von Neumann bottleneck (the data movement between processor and memory which is costly in terms of energy and latency), there is an increasing trend to move computing to the residence of data *i.e.* memory. In other words, the conventional method of storing data in a memory, fetching them and processing it in a dedicated processor before being written back to memory is being re-engineered. New computing paradigms are being explored which compute or process data at their location, if possible in the memory array itself and its peripherals (in–memory computing) or in a dedicated location near the memory (near–memory computing). In fact, **the computing architectures of future will not distinguish between memory unit and processing unit since they will be closely located or fused deliberately to avoid data movement**. No hardware engineer can afford to be ignorant of NVM since the applications of these memory devices are not just for storage, but also for various types of computing. Therefore, such a book introducing students and practising engineers to the fundamentals of ReRAMs is timely and needed. Finally, ReRAMs and other emerging memories are taught as part of a graduate-level course in semiconductor memories. No prior knowledge of conventional memories (SRAM, DRAM, Flash) is required to understand this book. However, a basic knowledge of electronic circuit design (Analog and Digital Integrated Circuit Design) is needed to understand the peripheral circuitry of ReRAM.

Erlangen, *John Reuben*
October 2023 *Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)*

* Vlasov, A. I.; Gudoshnikov, I. V.; Zhalnin, V. P.; Kadyr, A. T.; Shakhnov, V. A. 2020. Market for memristors and data mining memory structures for promising smart systems, Entrepreneurship and Sustainability Issues 8(2): 98-115. https://doi.org/10.9770/jesi.2020.8.2(6)

Acknowledgements

My journey with Resistive RAM (a type of memristor) started in the year 2017 in Israel. So, I start by thanking Prof. Shahar Kvatinsky, Technion- Israel Institute of Technology for his wonderful support. I was hosted at his $ASIC^2$ lab in The Andrew and Erna Viterbi Faculty of Electrical and Computer Engineering, Technion, Israel. Special thanks to Eric Herbelin, $ASIC^2$ lab manager for all his help in my time in Israel. Next, I moved to Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany. I was hosted by Prof. Dietmar Fey at the Chair of Computer Architecture. My heartfelt thanks to Prof. Fey for hosting me and also supporting me in all my research endeavors. I also thank the other members of Chair of Computer Architecture, FAU for their support. Special thanks to Edwin Aures, Lab Manager and the secretarial staff for all their help. Prof. Fey not only provided me with a good environment to conduct research but also nurtured me as a researcher. I sincerely thank him for taking time to review the entire manuscript of this book and giving constructive feedback to improve the manuscript.

I also thank Prof. Christian Wenger, IHP, Frankfurt Oder for collaborating on the research. Prof.Wenger shared his ReRAM expertise with us based on the Resistive RAM devices manufactured at IHP, Frankfurt Oder, Germany. Next, I thank the members of Memristec community in Germany for the various networking events and the collaborative works.

Leaving professional circle, it is time to be grateful to people in my personal life. I thank my family (wife and children) for supporting my research interests and moving with me to different countries and making the necessary adjustments. I thank my parents who decided to have a third child (that's me!) when the Indian norm was two children. I thank my brothers and sisters who added fun to my childhood, each one as different as my five fingers. I thank the God of the bible who sent his son Jesus Christ to be the savior of this world. This life, good health, knowledge and intelligence are gifts from God and I could not have written this book without His unseen hand.

Contents

Part I DEVICE, MODELING AND ARRAY STRUCTURE

1 Introduction 3
1.1 Do We Need New NVMs Beyond Flash Memory? 3
1.1.1 Non-Volatile Memory(NVM) 3
1.1.2 Flash Memory and its Challenges 3
1.2 Resistance-switching Memories 4
1.2.1 Resistive RAM: The Focus of This Book 6
1.3 General Scope of the Book: Intended Audience 6
1.4 Circuit Emphasis 7
References 8

2 Resistive RAM Device Characteristics 11
2.1 Resistive RAM and Memristor: A Clarification 11
2.2 Device Structure and Composition 12
2.3 Switching Characteristics 13
2.3.1 Switching Kinetics 15
2.3.2 Switching Energy 16
2.4 Endurance 17
2.4.1 How is it Measured and Reported? 17
2.4.2 Why is the Endurance Limited in ReRAM? 18
2.5 Retention 19
2.5.1 How is it Measured and Reported? 19
2.6 Variability 21
2.6.1 How is variability Measured and Reported? 22
2.6.2 How is Variability Controlled? 22

References ... 23

3 ReRAM Modeling ... 27
3.1 What is a Model for a Memory Device? ... 27
3.2 Approaches to ReRAM Device Modeling ... 28
3.3 Stanford-PKU ReRAM Model ... 30
3.3.1 Fitting Algorithm ... 31
3.4 Steps to Simulate a ReRAM Device using Stanford-PKU Model in Cadence Virtuoso ... 36
References ... 37

4 Array Structure and Memory Density ... 41
4.1 Memory Array ... 41
4.2 Resistive RAM Array Configurations ... 42
4.2.1 1R Configuration ... 43
4.2.2 1S–1R Configuration ... 44
4.2.3 1T–1R Configuration ... 46
4.3 Selectors for 1S–1R Array ... 47
4.3.1 Required Characteristics ... 47
4.3.2 Types of Selectors ... 48
4.4 Three Ways to Increase Memory Density ... 49
4.4.1 Device Scaling and Scalability ... 50
4.4.2 Multi-Level Cell (Multi-bit) ... 51
4.4.3 3D Stacking ... 52
4.4.3.1 Horizontally Stacked 3D ReRAM ... 53
4.4.3.2 Vertically Stacked 3D ReRAM ... 54
References ... 54

Part II PERIPHERAL CIRCUITRY

5 Circuits to Read from the ReRAM Array: Sense Amplifiers ... 61
5.1 Introduction ... 61
5.2 Read-out Circuits: Basic Principle of Sensing ... 61
5.3 Voltage-mode Sensing ... 63
5.4 Current-mode Sensing ... 66
5.5 Time-based Sensing ... 69
5.6 ReRAM technology-specific issues to be considered while sensing ... 71
5.6.1 Read-disturb ... 71
5.6.2 Random Telegraph Noise (RTN) ... 73

5.7 Sense Amplifier Design Example 73
References 75

6 Circuits to Program the ReRAM Array: WRITE Circuits 77
6.1 Introduction 77
6.2 Basic WRITE Circuit 78
6.2.1 Voltage Regulation during WRITE operation 80
6.3 Incremental Program and Verify Approach 82
6.4 WRITE Termination Approach 83
6.4.1 WRITE Termination by monitoring BL current . . . 84
6.4.2 WRITE Termination by monitoring SL current . . . 85
6.5 Writing into 1S-1R arrays 87
6.6 Issues faced while writing into ReRAM cell 89
6.6.1 Write disturb 89
6.6.2 IR drop 90
References 90

7 Row/Column Access Circuits and Multilevel Cell 93
7.1 Row Decoder: Circuit to Select a Row 93
7.1.1 WL driver 96
7.2 Column Multiplexers 96
7.2.1 Predecoding for large arrays 98
7.3 Multi-Level Cell(MLC): Some Directions for Writing and Reading 99
7.3.1 Writing Multiple States 99
7.3.2 Reading Multiple States 101
7.3.2.1 Sense Amplifier for multi-bit data 104
7.4 Memory Architecture: A Brief Introduction 105
7.4.1 Sub-array Design 106
References 108

Part III IN–MEMORY COMPUTING

8 In-Memory Computing: Applications of Resistive RAM Beyond Memory 113
8.1 Introduction 113
8.1.1 Von Neumann Bottleneck (The Memory Wall) 113
8.2 In-memory Arithmetic 116
8.2.1 Boolean Logic Gates in ReRAM array 116
8.2.1.1 NOR Logic Operation in Memory Array . . 118
8.2.1.2 Majority Logic Operation in Memory Array 119

8.2.2 In-memory Adders 123
8.2.2.1 One-bit Full Adder using In-memory NOR Gate 123
8.2.2.2 One-bit Full Adder using In-memory Majority Gate 124
8.2.2.3 Optimizing Latency of In-memory Adders: Some Directions 126
8.2.3 Eight-bit Parallel-prefix Adder in ReRAM array . . . 129
8.2.3.1 Mapping Methodology 130
8.3 Matrix Vector Multiplication(MVM) in ReRAM Array . . . 133
8.3.1 MVM by Bit Slicing 139
8.4 Conclusion and Future Outlook 143
References 144

Part I
DEVICE, MODELING AND ARRAY STRUCTURE

Chapter 1
Introduction

1.1 Do We Need New NVMs Beyond Flash Memory?

1.1.1 Non-Volatile Memory(NVM)

A non-volatile memory is a device which can retain data without being powered ON. From an electrical engineering perspective, it is a physical device which can store data without being continuously supplied with energy. It must be noted that writing into the device and reading from it does require energy. Energy may be required to write into the device and read from the device, but no energy is required to retain the stored data in the device. Once written, the data is unaltered and remains unaltered for a long period of time, typically many years. For how many years it can retain is defined by a characteristic of NVM called 'retention' and this will be discussed more in detail later.

1.1.2 Flash Memory and its Challenges

Flash memory is a type of non-volatile memory where the information is stored as a charge in the floating gate of a MOSFET structure *i.e.* in addition to a normal gate, the flash memory device has an floating gate to store charge. This presence or absence of charge functions as a memory [1]. Flash memory has been the de-facto standard for NVMs for decades. It has been used in our computers, USB sticks , mobile phones and almost all

electronic devices which need to retain some data without being powered ON.

In a textbook on emerging NVM like ReRAM, it is important to first lay the foundation by asking the question - why do we need **new** NVMs in the first place? In other words, is flash memory facing some limitations? Although flash is non-volatile and has been the de-facto standard for large storage, IRDS 2020 report [2] identifies the following inevitable limitations faced by flash memory:

1. Poor endurance* (10^3 to 10^5 erase cycles)
2. Modest retention (typically 10 years on a new device),
3. Long erase time (≈ms)
4. High operating voltage (15 V)

According to Waser et al. [3], flash memory suffers from low endurance, low write speed and high voltages required for WRITE operation. The read, write, and erase latency of flash memory technology have remained essentially constant for more than ten years, despite the technology's projections for greater scaling of density [4]. While the introduction of Multi-Level Cell (MLC) flash devices extended their storage capacities by a small integral factor (2–4), the combination of scaling and MLC have resulted in the degradation of both retention time and endurance, two parameters critical for storage applications [2]. All these challenges has invigorated research in the last two decades for a potential replacement for flash memory.

1.2 Resistance-switching Memories

A resistance-switching memory is any two-terminal resistor device whose resistance can be electrically altered in two stables states [5]. Under voltage/current stress, the resistance can be switched between a Low Resistance State (LRS) and a High Resistance State (HRS) (hence the name resistance-switching). The word 'memristor' is also used by researchers to refer to a resistive-switching device since such a device is basically a 'resistor' with a 'memory'. The word 'memristor' is a broad word and is used to refer to any resistance-switching memory regardless of the underlying device structure

* Endurance denotes the number of times that the device can be switched between two stable states while maintaining enough resistance ratio between them. This will be discussed in detail in Chapter 2

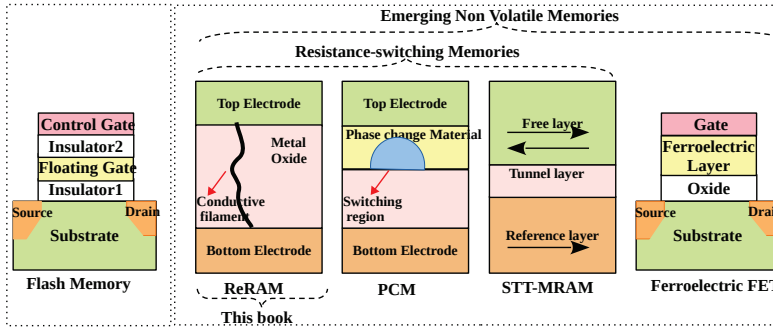


Fig. 1.1 Flash memory (left) and emerging NVMs (right) seeking to replace flash memory, at least partially. ReRAM, PCM and STT-MRAM are resistance-switching memories since the data is stored as resistance. Ferroelectric FET is a type of ferroelectric memory where the data is stored as the threshold voltage of the MOSFET, similar to flash memory (Ferroelectric RAM (FeRAM) and Ferroelectric Tunnel Junction(FTJ) are other types of ferroelectric memories not shown here). The book focuses only on Resistive RAM.

or the mechanism which causes the change in resistance. To differentiate between different types of resistance-switching memories, we need to look at their structure and the accompanying physical mechanism which causes the resistance change. Following this approach, they can be classified as follows:

1. Resistive Random Access Memory (ReRAM) device is a Metal-Insulator-Metal structure where a conductive filament is created (LRS) or broken (HRS) in the insulator. The insulator is usually a transition metal oxide (OxRAM) or an electrolyte (Conductive Bridge RAM)
2. Phase Change Memory (PCM) device is a Metal-Active Material-Metal structure where the active material is a chalcogenide phase-change material which is either in amorphous (HRS) or crystalline state (LRS)
3. Spin Transfer Torque-Magnetic RAM (STT-MRAM) is a Free layer-Tunnel Layer-Reference layer structure where the magnetic polarization of the Reference layer is fixed while that of the free layer can be programmed to be either in the same direction (parallel, LRS) or opposite direction (anti-parallel, HRS)

Resistance switching memories (ReRAM, PCM, STT-MRAM), ferroelectric memories (FTJ, FeFET, FeRAM), macromolecular(polymer) memory, mott memory and DNA memory are some of the emerging memories and each of them are in different stages of maturity. IRDS 2022 report

[6] identifies STT-MRAM, PCM, ReRAM and FeRAM as prototypical memories since they are available as prototypes and even commercially available for niche applications.

1.2.1 Resistive RAM: The Focus of This Book

ReRAM, STT-MRAM and PCM are prominent resistance-switching memories in the early stages of commercialization. All these resistance-switching memories have their own merits and demerits and a comprehensive comparison is not possible since they are in nascency. This book will focus on ReRAM, an emerging NVM having a greater interest and potential compared to other competing NVMs mentioned above. This has been recognised by various researchers and experts in semiconductor memory industry and consequently these memories are in the early stages of manufacturing. Fujitsu manufactures 12 Mbit ReRAM [7]. Infineon is using TSMC's ReRAM in its aurix microcontrollers [8]. Crossbar Inc [9] and Weebit Nano [10] license ReRAM IPs and many other semiconductor memory manufacturers are also entering the market. According to Yole's NVM market intelligence report, emerging NVM market is expected to reach $2.7 billion by 2028, with ReRAM expected to account for more than a third of that [11].

1.3 General Scope of the Book: Intended Audience

Although the book focuses on ReRAM, there is a wider scope for this book in the context of NVM. As a reader or expert in this field of emerging NVM will agree, the basic principles of NVM discussed in the context of ReRAM are applicable for other two-terminal NVM with little or no modifications. For *e.g.*, the chapter on Array Structure discusses different ways in which ReRAM devices are organised into arrays– as single devices (1R) or with a selector(1S-1R) or with a transistor (1T-1R). STT-MRAM is also a two-terminal device which is commonly integrated in a 1T-1R architecture with a magnetic RAM integrated on top of the drain of the transistor instead of the ReRAM. Similarly, the peripheral circuitry (READ and WRITE circuits, row and column decoders) are also very similar and can be adapted to any resistance-switching memory. Hence, this book will be

useful for anyone who wants an overview of emerging NVM in addition to those desiring a detailed view of ReRAM technology.

This book caters to various professions. First, it can be used as a textbook for graduate students of electrical/electronics/computer engineering. Most of the universities have a module on memory as part of computer architecture course. The module covers the basics of SRAM, DRAM and flash memory. However, this was the case when flash memory dominated the NVM market for storage class memory. A course on emerging NVM is already beginning to be offered as a separate course in some universities due to the attention these memories have got in recent past. This book can be used as a textbook for an introductory course on ReRAM or used as part of a larger course on emerging NVM. Secondly, practising semiconductor engineers in the field of memory IP development will benefit from the exhaustive coverage of peripheral circuitry. Thirdly, this book will be useful for researchers entering the field of in-memory computing, neuromorphic computing and AI hardware [12, 13], all of which use array of some resistance switching memory as the substrate for processing.

1.4 Circuit Emphasis

Any discussion on electronics can be at different levels of abstraction-device or circuit or system. Any textbook on integrated circuits must make the emphasis clear to the audience. A book on CMOS integrated circuits can place much emphasis at the device level discussing the transistor device physics and the various phenomenon occurring inside that single transistor. Another book can have an emphasis on circuit design - the art of designing circuits using the transistors. A book with system emphasis can elaborate how complex systems like System-on-Chip (SoC) can be designed by stitching together many IP cores. The emphasis of this book is at the abstraction of circuit. In Part I, the author discusses the basic device structure, characteristics, modeling and how these devices are integrated to form a memory array. However, the switching mechanism and the accompanying resistance change are not discussed at the ion level. Neither is device physics the emphasis of the book[†]. This book views the memory

[†] The subject of ReRAM device physics and switching mechanism at the device level is well presented in other books. *e.g. Advances in Non-Volatile Memory and Storage Technology*, Woodhead Publishing, 2019, ISBN 9780081025840; *Resistive*

device at the circuit level and gives the minimum device understanding a memory chip designer needs. Given a ReRAM device and its characteristics, how can one design a Random Access Memory(RAM) using it? Part II discusses peripheral circuitry- how to design circuits to read from and write into the memory array? How to design the row and column access control circuits? Part III gives an introduction to the field of in-memory computing. Again, the emphasis is on circuit design- how can ReRAM devices and CMOS circuits be used together to achieve simple computations like addition and matrix vector multiplication? In short, the book tries to answer the questions of a memory circuit designer and not of a memory device engineer.

References

[1] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," *Proceedings of the IEEE*, vol. 91, no. 4, pp. 489–502, 2003.

[2] International roadmap for devices and systems: Beyond cmos. [Online]. Available: https://irds.ieee.org/editions/2020/beyond-cmos

[3] R. Waser, R. Dittmann, G. Staikov, and K. Szot, "Redox-based resistive switching memories – nanoionic mechanisms, prospects, and challenges," *Advanced Materials*, vol. 21, no. 25-26, pp. 2632–2663, 2009.

[4] L. M. Grupp, A. M. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. H. Siegel, and J. K. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 42. New York, NY, USA: Association for Computing Machinery, 2009. doi: 10.1145/1669112.1669118. ISBN 9781605587981 p. 24–33. [Online]. Available: https://doi.org/10.1145/1669112.1669118

[5] D. J. Wouters, R. Waser, and M. Wuttig, "Phase-change and redox-based resistive switching memories," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1274–1288, 2015.

[6] Beyond cmos and emerging materials integration, irds 2022 report. [Online]. Available: https://irds.ieee.org/editions/2022/irds-2022-beyond-cmos-and-emerging-research-materials

[7] (2023) Fujitsu semiconductor memory solution data sheet. [Online]. Available: https://www.fujitsu.com/jp/group/fsm/en/documents/products/reram/lineup/MB85AS12MT-DS501-00071-1v1-E13.pdf (Accessed 2023-10-30).

[8] (2022) Infineon and tsmc to introduce rram technology for automotive aurix tc4x product family. [Online].

Switching: From Fundamentals of Nanoionic Redox Processes to Memristive Device Applications, Wiley-VCH Verlag, 2016, ISBN 9783527334179

Available: https://www.infineon.com/cms/en/about-infineon/press/market-news/2022/INFATV202211-031.html (Accessed 2023-10-30).

[9] (2022) Rethink reram for use as ftp or otp memory. [Online]. Available: https://www.crossbar-inc.com/products/ftp-otp-memory/ (Accessed 2023-10-30).

[10] (2023) Non-volatile memory (nvm) without compromise. [Online]. Available: https://www.weebit-nano.com/technology/overview/ (Accessed 2023-10-30).

[11] (2023) Emerging non-volatile memory 2023. [Online]. Available: https://www.yolegroup.com/product/report/emerging-non-volatile-memory-2023/ (Accessed 2023-10-30).

[12] *Applications of Emerging Memory Technology Beyond Storage*, 1st ed., ser. Springer Series in Advanced Microelectronics, 63. Singapore: Springer Singapore, 2020. ISBN 981-13-8379-0

[13] K. Ishimaru, "Future of non-volatile memory -from storage to computing-," in *2019 IEEE International Electron Devices Meeting (IEDM)*, 2019. doi: 10.1109/IEDM19573.2019.8993609 pp. 1.3.1–1.3.6.

Chapter 2
Resistive RAM Device Characteristics

2.1 Resistive RAM and Memristor: A Clarification

A memristor is a contraction for the combination of memory and resistor *i.e.* a device which is capable of remembering its resistance. The word 'memristor' was coined by Leon Chua in his 1971 paper- *'Memristor-The missing circuit element'* [1]. In this paper, the author stated that such a device is yet to be discovered, but theoretical investigations prove the existence of such a device. Since the word memristor does not say anything about the the device structure or the mechanism which leads to change in resistance, it can be easily misunderstood*. Furthermore, the word memristor is broad and refers to any device which can change its resistance in response to voltage/current stimulation. In fact, all two-terminal resistance switching memories are memristors, regardless of the device material and physical operating mechanisms [2]. Hence, the word memristor is very broad and is agnostic to the underlying device structure or the mechanism which causes the change in resistance.

The word Resistive RAM, on the other hand, is more specific. Firstly, the word ReRAM refers to a two-terminal device in which the data is stored as resistance. Secondly, in a ReRAM device, the change in resistance is due to the motion of electrically active ionic defects (*e.g.* oxygen vacancies or metal cations). This movement of ionic defects results in the formation or rupture of a conductive filament in the device. What makes a resistance switching memory to be classified as ReRAM is the

* In technical literature, the word memristor and Resistive RAM were often used interchangeably and this causes a confusion for a person entering this field.

cause of resistance change *i.e.* movement of ionic defects leading to the formation of a conductive filament in the insulator (HRS → LRS) and its rupture (LRS → HRS) under appropriate voltage stimulus. Note that STT-MRAM and PCM also exhibit change in resistance under voltage stress, but they are not classified as ReRAM since the change in resistance is not due to the formation of conductive filament. On the contrary, in PCM, the change in resistance is due to change in the material property (from amorphous to crystalline) and in STT-MRAM, the change in resistance is due to polarization alignment between the free layer and the reference layer. Hence, STT-MRAM and PCM can be classified as memristors but not as Resistive RAMs. To summarize, the word memristor is a broad term referring to resistance switching devices and is agnostic to device structure and the physical mechanism which causes the resistance change. The word 'Resistive RAM' or 'ReRAM' specifically refers to devices which store data as resistance **and** the accompanying change in resistance is due to the movement of electrically active ionic defects (which results in formation or rupture of conductive filament). Finally, the suffix 'RAM' in ReRAM is used in the same sense as it is used in SRAM and DRAM *i.e.* it is a memory which permits READ and WRITE operations and the access time for any location in the memory array is the same.

2.2 Device Structure and Composition

Resistive RAM device is a 'Top Electrode–Switching Material–Bottom Electrode' structure, as shown in Fig 2.1. The **state** of the ReRAM device is its resistance *i.e.* the data is stored as resistance and its resistance does not change even after the memory chip is left without any source of electrical power. The resistance can be changed by the application of voltage between the top and bottom electrodes. Under such a voltage stress, a conductive filament is formed or ruptured in the switching material, which leads to change in resistance. Two types of Resistive RAM are distinguished based on the switching material- OxRAM, if the switching material between the two electrodes is a binary metal oxide or Conductive Bridging RAM, if the switching material is oxide or electrolyte (with the top electrode being an electrochemically active metal) [3]. OxRAM and Conductive bridging RAM not only differ in their switching material, but also in the chemical process by which the conductive filament is formed. This is discussed in the next section. Another important characteristic of ReRAM

technology is that the switching material and the metal electrodes are not standardized. Different oxides like HfO_2, TaO_2, SiO_2, TiO_2, Ta_2O_5 [4, 5] have been investigated as switching material and different metals like platinum, titanium, aluminium, carbon have been investigated as electrodes [6, 7, 5]. Consequently, ReRAM device characteristics vary significantly between memory manufacturers according to the switching material and its thickness, electrodes used *etc.*

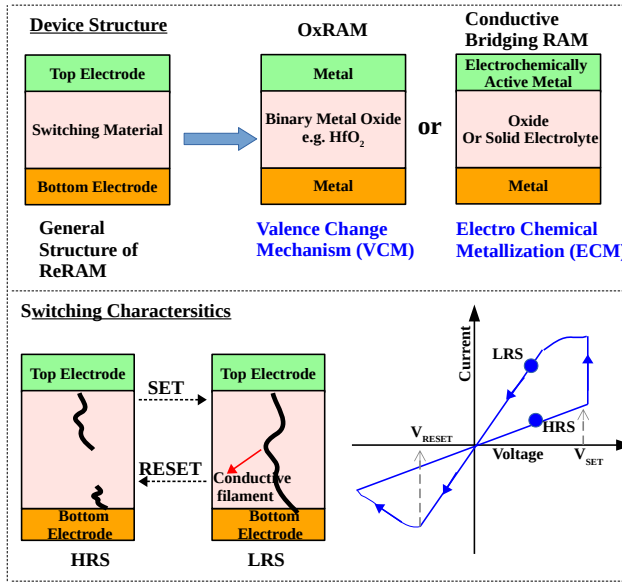


Fig. 2.1 Two terminal structure of ReRAM: OxRAM and Conductive Bridging RAM differ in the composition of the electrodes and switching material and the mechanism by which the filament is formed. In OxRAM, it is by Valence Change Mechanism and in Conductive Bridging RAM it is by Electro Chemical Metallization

2.3 Switching Characteristics

As depicted in Fig. 2.1, the change in resistance in a ReRAM device is due to the formation or rupture of a conducive filament between top and bottom electrodes. The transition from HRS to LRS is called the 'SET'

rram as synaptic weight element," in *2016 IEEE Silicon Nanoelectronics Workshop (SNW)*, pp. 50–51, 2016.

[34] A. Chen, "Utilizing the variability of resistive random access memory to implement reconfigurable physical unclonable functions," *IEEE Electron Device Letters*, vol. 36, no. 2, pp. 138–140, 2015.

[35] R. Degraeve, A. Fantini, N. Raghavan, L. Goux, S. Clima, B. Govoreanu, A. Belmonte, D. Linten, and M. Jurczak, "Causes and consequences of the stochastic aspect of filamentary rram," *Microelectronic Engineering*, vol. 147, pp. 171 – 175, 2015. Insulating Films on Semiconductors 2015.

[36] D. Ielmini and V. Milo, "Physics-based modeling approaches of resistive switching devices for memory and in-memory computing applications," *J. Comput. Electron.*, vol. 16, pp. 1121–1143, Dec. 2017.

[37] V. Y. Zhuo, Y. Jiang, R. Zhao, L. P. Shi, Y. Yang, T. C. Chong, and J. Robertson, "Improved switching uniformity and low-voltage operation in taox-based rram using ge reactive layer," *IEEE Electron Device Letters*, vol. 34, pp. 1130–1132, Sept 2013.

[38] E. Ambrosi, A. Bricalli, M. Laudato, and D. Ielmini, "Impact of oxide and electrode materials on the switching characteristics of oxide reram devices," *Faraday Discuss.*, pp. –, 2018.

[39] G. C. Adam, A. Khiat, and T. Prodromakis, "Challenges hindering memristive neuromorphic hardware from going mainstream," *Nature Communications*, vol. 9, 2018.

[40] E. Pérez, C. Zambelli, M. K. Mahadevaiah, P. Olivo, and C. Wenger, "Toward reliable multi-level operation in rram arrays: Improving post-algorithm stability and assessing endurance/data retention," *IEEE Journal of the Electron Devices Society*, vol. 7, pp. 740–747, 2019.

[41] J. B. Roldán, E. Miranda, D. Maldonado, A. N. Mikhaylov, N. V. Agudov, A. A. Dubkov, M. N. Koryazhkina, M. B. González, M. A. Villena, S. Poblador, M. Saludes-Tapia, R. Picos, F. Jiménez-Molinos, S. G. Stavrinides, E. Salvador, F. J. Alonso, F. Campabadal, B. Spagnolo, M. Lanza, and L. O. Chua, "Variability in resistive memories," *Advanced Intelligent Systems*, vol. n/a, no. n/a, p. 2200338.

[42] E. Shahrabi, C. Giovinazzo, J. Sandrini, and Y. Leblebici, "The key impact of incorporated al2o3 barrier layer on w-based reram switching performance," in *2018 14th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, pp. 69–72, July 2018.

[43] A. Hardtdegen, C. La Torre, F. Cüppers, S. Menzel, R. Waser, and S. Hoffmann-Eifert, "Improved switching stability and the effect of an internal series resistor in hfo2/tioxbilayer reram cells," *IEEE Transactions on Electron Devices*, vol. 65, pp. 3229–3236, Aug 2018.

[44] I. Sanchez Esqueda, H. Zhao, and H. Wang, "Efficient learning and crossbar operations with atomically-thin 2-D material compound synapses," *Journal of Applied Physics*, vol. 124, 10 2018. 152133.

Chapter 3
ReRAM Modeling

3.1 What is a Model for a Memory Device?

Roughly speaking, **modeling** refers to the way in which the physical switching phenomenon is imitated by a computer (software program) so that circuit designers can understand the working of a device and other larger circuits (using those devices) without fabricating the device. Because fabricating the memory array and the circuits around it (to read and write) is costly, researchers depend on models to get an initial feel or a feasibility prediction of a memory technology. This was the same case with CMOS technology and SPICE models were developed to design and simulate circuits to study the feasibility of a particular chip or microprocessor before going for large scale production. In the same way, once a memory technology attains reasonable maturity, models are developed to study the feasibility of a memory technology for commercial applications. Consider a 256×256 array of memory cells along with peripheral circuits to write and read from the memory array. If a model for a ReRAM device is developed, the memory array can be simulated. With SPICE models for CMOS peripheral circuits, reading and writing to the memory array can be verified by simulation. In this manner, the feasibility of the entire memory system can be studied (WRITE energy consumption, READ latency etc). This will aid in the progress and commercialization of a particular memory technology. Therefore, it is important for engineers/researchers to understand how a memory device is modeled so that circuit simulations can be performed using the models.

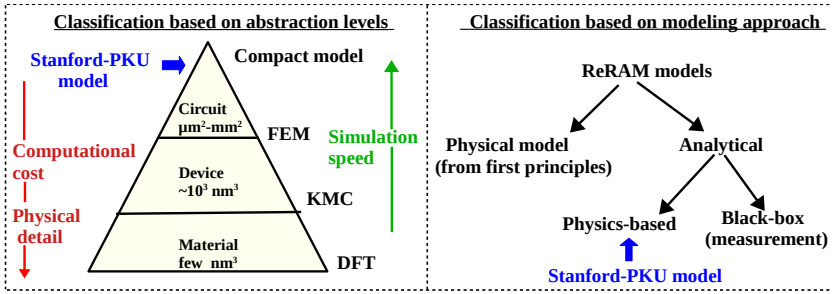


Fig. 3.1 Based on degree of detail, a model for ReRAM can range from atomistic (Density Functional Theory (DFT)) to Kinetic Monte Carlo (KMC) to Finite Element Method (FEM) to compact model [1]. From another perspective, a model for ReRAM can be derived from first principles or simplifications.

3.2 Approaches to ReRAM Device Modeling

The resistive switching phenomenon (the change in resistance of the ReRAM cell in response to voltage stress) has been modeled by various groups using different approaches [2, 1]. A classification of ReRAM models based on the degree of detail is presented in [1] (Fig. 3.1). While atomistic models capture intricate details like ion/atom diffusion and migration mechanisms at atomistic scale (few nm^3), compact models capture macroscopic details like geometry of the conductive filament and temperature. Based on the modeling approach, they can be classified as physical models and analytical models (Fig. 3.1). Physical models try to model resistive switching behavior from first principles, *i.e.* by modeling the fundamental cause for change in resistance, from the perspective of device-physics. On the other hand, analytical models are formulated as mathematical equations, which match the device's experimentally-observed behaviour. The physical principles approach and analytical approach are two extremes, and in reality, a model for ReRAM is a compromise between these two extremes [3]. Analytical approaches can be further classified as being either physics-based or black-box. In a physics-based analytical model, the physics of resistive switching is modeled by appropriate equations *e.g.* a physics-based analytical model may simplify resistive switching to the formation and rupture of conductive filament under voltage stress. In a black-box model, a measurement approach is followed and the approach is agnostic to the device structure or switching mechanism *e.g.* a device is subjected to different stimulus and its response is studied; then

[12] H. Li, Z. Jiang, P. Huang, Y. Wu, H. . Chen, B. Gao, X. Y. Liu, J. F. Kang, and H. . P. Wong, "Variation-aware, reliability-emphasized design and optimization of rram using spice model," in *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1425–1430, March 2015.

[13] L. Chua, "Device modeling via nonlinear circuit elements," *IEEE Transactions on Circuits and Systems*, vol. 27, pp. 1014–1044, November 1980.

[14] X. Guan, S. Yu, and H. . P. Wong, "A spice compact model of metal oxide resistive switching memory with variations," *IEEE Electron Device Letters*, vol. 33, pp. 1405–1407, Oct 2012.

[15] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H. . P. Wong, "A neuromorphic visual system using rram synaptic devices with sub-pj energy and tolerance to variability: Experimental characterization and large-scale modeling," in *2012 International Electron Devices Meeting*, pp. 10.4.1–10.4.4, Dec 2012.

[16] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, and H. . P. Wong, "Verilog-a compact model for oxide-based resistive random access memory (rram)," in *2014 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, pp. 41–44, Sept 2014.

[17] J. Reuben, D. Fey, and C. Wenger, "A modeling methodology for resistive ram based on stanford-pku model with extended multilevel capability," *IEEE Transactions on Nanotechnology*, vol. 18, pp. 647–656, 2019.

[18] J. Sandrini, "Fabrication, characterization and integration of resistive random access memories," p. 241, 2017.

[19] E. Pérez, A. Grossi, C. Zambelli, P. Olivo, R. Roelofs, and C. Wenger, "Reduction of the cell-to-cell variability in hf1-xalxoy based rram arrays by using program algorithms," *IEEE Electron Device Letters*, vol. 38, no. 2, pp. 175–178, 2017.

[20] V. Y.-Q. Zhuo, Y. Jiang, R. Zhao, L. P. Shi, Y. Yang, T. C. Chong, and J. Robertson, "Improved switching uniformity and low-voltage operation in tao_x-based rram using ge reactive layer," *IEEE Electron Device Letters*, vol. 34, no. 9, pp. 1130–1132, 2013.

[21] E. Ambrosi, A. Bricalli, M. Laudato, and D. Ielmini, "Impact of oxide and electrode materials on the switching characteristics of oxide reram devices," *Faraday Discuss.*, vol. 213, pp. 87–98, 2019.

[22] G. González-Cordero, J. B. Roldan, F. Jiménez-Molinos, J. Suñé, S. Long, and M. Liu, "A new compact model for bipolar rrams based on truncated-cone conductive filaments—a verilog-a approach," *Semiconductor Science and Technology*, vol. 31, no. 11, p. 115013, 2016.

[23] Y. Huang, Z. Shen, Y. Wu, M. Xie, Y. Hu, S. Zhang, X. Shi, and H. Zeng, "Cuo/zno memristors via oxygen or metal migration controlled by electrodes," *AIP Advances*, vol. 6, no. 2, p. 025018, 2016.

[24] A. Bricalli, E. Ambrosi, M. Laudato, M. Maestro, R. Rodriguez, and D. Ielmini, "Resistive switching device technology based on silicon oxide for improved on–off ratio—part i: Memory devices," *IEEE Transactions on Electron Devices*, vol. 65, pp. 115–121, Jan 2018.

[25] J. Reuben, M. Biglari, and D. Fey, "Incorporating variability of resistive ram in circuit simulations using the stanford–pku model," *IEEE Transactions on Nanotechnology*, vol. 19, pp. 508–518, 2020.

Chapter 4
Array Structure and Memory Density

4.1 Memory Array

The non-volatile memory cells explored in the previous chapters have to be arranged in some fashion to be usable for storing and retrieving data in an organised manner. Conventional memories like SRAM and DRAM were always arranged in arrays. We define the memory array as a regular array of memory cells to store data, replicated in two dimensions– rows and columns. As depicted in Fig. 4.1 (a), the memory array of 8 rows and 8 columns can store 64 bits of information, where each memory cell stores a bit. We can call this the logical view. Just like a NAND gate is represented by a symbol in a circuit diagram but the detailed structure is made of 4 transistors (2 PMOS and 2 NMOS transistors), the structural view of the memory array is different from this logical view. The logical view is a more abstract view than the structural view. Therefore the logical view is the same but the structural view is different for different memory technologies. To illustrate, Fig. 4.1 (b) and (c) depict the detailed structure of 4×4 DRAM and SRAM array, respectively. We can call this the structural view. In practise, the structural view is important for circuit designers while those working higher at a system level will only 'see' the memory as arrangement of bits of information (in rows and columns).

As depicted, the structural view is dependent on the memory technology. For DRAM, each memory cell is a capacitor with an access transistor. But for SRAM, each memory cell is back-to-back connected inverters with two access transistors. Not only is the structural view different for different memory technologies, the way in which a bit of information is stored is also significantly different in different memory

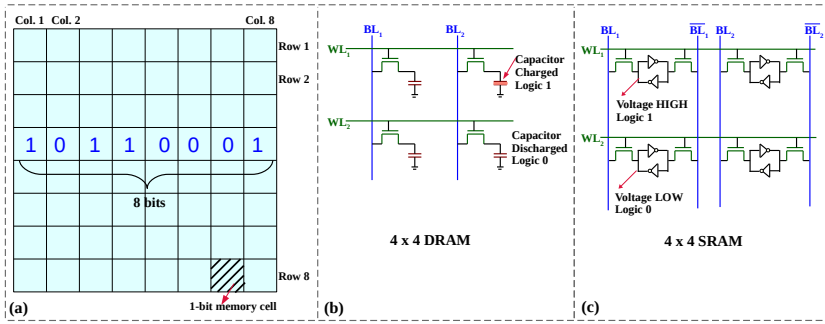


Fig. 4.1 (a) Logical view of the memory array: Each bit is just a location in the memory array. (b) Structural view of DRAM: Each bit of information is stored as a charge in the capacitor (c) Structural view of SRAM: Each bit is charge at the input of the back-to-back connected inverters

technologies. In DRAM, the information is stored as the charge in a capacitor while in SRAM the information is stored as a charge, which results in a voltage at the cross-coupled inverter. Another factor which is obvious from the structural view is the area occupied by a memory array in silicon. Clearly, 256×256 DRAM array will not occupy the same area as 256×256 SRAM array due to the difference in the area occupied by a single memory cell (Area of single DRAM cell is $6F^2$ while that of SRAM cell is $200F^2$ [1], where F is the feature size, explained in the next section). We have elaborated on the logical view and the structural view just to clarify the basic terminologies of the conventional memories. In the following sections, the commonly pursued array structures for ReRAMs will be presented.

4.2 Resistive RAM Array Configurations

The 1R configuration (1 **R**eRAM device forms a memory cell) is simply the ReRAM device fabricated between two metal layers. The **1S–1R** configuration is the same as 1R with a selector device fabricated in series with the ReRAM (1 **S**elector and 1 **R**eRAM device form a memory cell). **1T–1R** is the configuration where each memory cell is fabricated with a transistor to enable access to the memory device (1 **T**ransistor and 1 **R**eRAM device form a memory cell).

transistors," in *2021 IEEE International Memory Workshop (IMW)*, pp. 1–4, 2021.

[19] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao, S. Joshi, H. Wu, H.-S. P. Wong, and G. Cauwenberghs, "A compute-in-memory chip based on resistive random-access memory," *Nature*, vol. 608, no. 7923.

[20] G. W. Burr, R. S. Shenoy, and H. Hwang, "Select device concepts for crossbar arrays," in *Resistive Switching*, ch. 22, pp. 623–660, John Wiley & Sons, Ltd, 2016.

[21] A. Chen, "Memory select devices," in *Emerging Nanoelectronic Devices*, ch. 12, pp. 227–245, John Wiley & Sons, Ltd, 2014.

[22] A. Chen, "A review of emerging non-volatile memory (NVM) technologies and applications," *Solid State Electronics*, vol. 125, pp. 25–38, Nov. 2016.

[23] Y. S. Chen, H. Y. Lee, P. S. Chen, P. Y. Gu, C. W. Chen, W. P. Lin, W. H. Liu, Y. Y. Hsu, S. S. Sheu, P. C. Chiang, W. S. Chen, F. T. Chen, C. H. Lien, and M.-J. Tsai, "Highly scalable hafnium oxide memory with improvements of resistive distribution and read disturb immunity," in *2009 IEEE International Electron Devices Meeting (IEDM)*, pp. 1–4, 2009.

[24] B. Govoreanu, G. Kar, Y.-Y. Chen, V. Paraschiv, S. Kubicek, A. Fantini, I. Radu, L. Goux, S. Clima, R. Degraeve, N. Jossart, O. Richard, T. Vandeweyer, K. Seo, P. Hendrickx, G. Pourtois, H. Bender, L. Altimime, D. Wouters, J. Kittl, and M. Jurczak, "10×10nm2 hf/hfox crossbar resistive ram with excellent performance, reliability and low-energy operation," in *2011 International Electron Devices Meeting*, pp. 31.6.1–31.6.4, 2011.

[25] D. Ielmini, "Resistive switching memories based on metal oxides: mechanisms, reliability and scaling," *Semiconductor Science Technology*, vol. 31, p. 063002, June 2016.

[26] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai, "Metal–oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.

[27] A. Prakash and H. Hwang, "Multilevel cell storage and resistance variability in resistive random access memory," *Physical Sciences Reviews*, vol. 1, no. 6, pp. –, 2016.

[28] J. Reuben, D. Fey, and C. Wenger, "A modeling methodology for resistive ram based on stanford-pku model with extended multilevel capability," *IEEE Transactions on Nanotechnology*, vol. 18, pp. 647–656, 2019.

[29] A. Prakash, J. Park, J. Song, J. Woo, E. Cha, and H. Hwang, "Demonstration of low power 3-bit multilevel cell characteristics in a taox-based rram by stack engineering," *IEEE Electron Device Letters*, vol. 36, pp. 32–34, Jan 2015.

[30] U. Russo, D. Kamalanathan, D. Ielmini, A. L. Lacaita, and M. N. Kozicki, "Study of multilevel programming in programmable metallization cell (pmc) memory," *IEEE Transactions on Electron Devices*, vol. 56, pp. 1040–1047, May 2009.

[31] S. Yu, Y. Wu, and H.-S. P. Wong, "Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory," *Applied Physics Letters*, vol. 98, p. 103514, 03 2011.

[32] A. Nitayama and H. Aochi, "Bit cost scalable (bics) technology for future ultra high density memories," in *2013 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, pp. 1–2, 2013.

[33] Y. Deng, H.-Y. Chen, B. Gao, S. Yu, S.-C. Wu, L. Zhao, B. Chen, Z. Jiang, X. Liu, T.-H. Hou, Y. Nishi, J. Kang, and H.-S. P. Wong, "Design and optimization methodology for 3d rram arrays," in *2013 IEEE International Electron Devices Meeting*, pp. 25.7.1–25.7.4, 2013.

[34] B. Hudec, C.-C. Chang, I.-T. Wang, K. Fröhlich, and T.-H. Hou, "Three dimensional integration of rerams," in *2018 IEEE 18th International Conference on Nanotechnology (IEEE-NANO)*, pp. 1–2, 2018.

[35] K. Son, K. Cho, S. Kim, G. Park, K. Song, and J. Kim, "Modeling and signal integrity analysis of 3d xpoint memory cells and interconnections with memory size variations during read operation," in *2018 IEEE Symposium on Electromagnetic Compatibility, Signal Integrity and Power Integrity*, pp. 223–227, 2018.

[36] D. Waddington, M. Kunitomi, C. Dickey, S. Rao, A. Abboud, and J. Tran, "Evaluation of intel 3d-xpoint nvdimm technology for memory-intensive genomic workloads," in *Proceedings of the International Symposium on Memory Systems*, MEMSYS '19, (New York, NY, USA), p. 277–287, Association for Computing Machinery, 2019.

[37] L. Zhang, S. Cosemans, D. J. Wouters, B. Govoreanu, G. Groeseneken, and M. Jurczak, "Analysis of vertical cross-point resistive memory (vrram) for 3d rram design," in *2013 5th IEEE International Memory Workshop*, pp. 155–158, 2013.

[38] D. J. Wouters, R. Waser, and M. Wuttig, "Phase-change and redox-based resistive switching memories," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1274–1288, 2015.

[39] B. Hudec, C.-W. Hsu, I.-T. Wang, W.-L. Lai, C.-C. Chang, T.-C. Wang, K. Frohlich, C. H. Ho, C.-H. Lin, and T. Hou, "3d resistive ram cell design for high-density storage class memory—a review," *Science China Information Sciences*, vol. 59, pp. 1–21, 2016.

Part II
PERIPHERAL CIRCUITRY

Chapter 5
Circuits to Read from the ReRAM Array: Sense Amplifiers

5.1 Introduction

The peripheral circuitry consists of circuits around the memory array which are usually constructed using CMOS technology. By 'peripheral circuitry', we mean the circuits used to write data (*i.e.* to program the memory cells), to read out data (*i.e.* sense amplifiers used to convert the state of the memory cell to a voltage for further processing), the circuits used to select a row and a column (decoders/multiplexers) and any other additional circuits which are crucial while accessing the memory array (*e.g.* error detection/correction circuits). In short, the peripheral circuits form a channel through which the data in electronic form (voltage) is moved into the memory array (to be stored as resistance) and moved out of the memory array (from resistance to a CMOS-compatible voltage for further processing).

5.2 Read-out Circuits: Basic Principle of Sensing

As stated, the read-out circuits mainly serve the purpose of converting the state of the memory cell to a voltage which can be processed further. Traditionally, processing of data has been performed outside the memory*

* In this book, the memory is used only as a storage medium so far. The in-memory computing paradigm seeks to compute/process data at the location of the data *i.e.* the memory array itself. This is an important development in this field and this is discussed in Part III. Even if data is processed in the memory array,

(NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC), pp. 1–6, 2019.

[13] J. Reuben and S. Pechmann, "Accelerated addition in resistive ram array using parallel-friendly majority gates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1108–1121, 2021.

[14] X. Zhang, B.-K. An, and T. T.-H. Kim, "A robust time-based multi-level sensing circuit for resistive memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 1, pp. 340–352, 2023.

[15] N. Raghavan, "Statistics of disturb events in oxram devices — a phenomenological model," in *2017 IEEE International Reliability Physics Symposium (IRPS)*, pp. PM–3.1–PM–3.7, 2017.

[16] M. Zhao, B. Gao, J. Tang, H. Qian, and H. Wu, "Reliability of analog resistive switching memory for neuromorphic computing," *Applied Physics Reviews*, vol. 7, p. 011301, 01 2020.

[17] W. Shim, Y. Luo, J.-S. Seo, and S. Yu, "Investigation of read disturb and bipolar read scheme on multilevel rram-based deep learning inference engine," *IEEE Transactions on Electron Devices*, vol. 67, no. 6, pp. 2318–2323, 2020.

[18] A. Prakash and H. Hwang, "Multilevel cell storage and resistance variability in resistive random access memory," *Physical Sciences Reviews*, vol. 1, no. 6, pp. –, 2016.

[19] S. Dongaonkar, M. D. Giles, A. Kornfeld, B. Grossnickle, and J. Yoon, "Random telegraph noise (rtn) in 14nm logic technology: High volume data extraction and analysis," in *2016 IEEE Symposium on VLSI Technology*, pp. 1–2, 2016.

[20] J. Kim, H. Nili, N. D. Truong, T. Ahmed, J. Yang, D. S. Jeong, S. Sriram, D. C. Ranasinghe, S. Ippolito, H. Chun, and O. Kavehei, "Nano-intrinsic true random number generation: A device to data study," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2615–2626, 2019.

[21] S. Ambrogio, S. Balatti, A. Cubeta, A. Calderoni, N. Ramaswamy, and D. Ielmini, "Statistical fluctuations in hfox resistive-switching memory: Part ii—random telegraph noise," *IEEE Transactions on Electron Devices*, vol. 61, no. 8, pp. 2920–2927, 2014.

[22] F. M. Puglisi, P. Pavan, L. Larcher, and A. Padovani, "Analysis of rtn and cycling variability in hfo2 rram devices in lrs," in *2014 44th European Solid State Device Research Conference (ESSDERC)*, pp. 246–249, 2014.

[23] S. Pechmann, T. Mai, M. Völkel, M. K. Mahadevaiah, E. Perez, E. Perez-Bosch Quesada, M. Reichenbach, C. Wenger, and A. Hagelauer, "A versatile, voltage-pulse based read and programming circuit for multi-level rram cells," *Electronics*, vol. 10, no. 5, 2021.

Chapter 6
Circuits to Program the ReRAM Array: WRITE Circuits

6.1 Introduction

The reader might recall from Chapter 2 that the cell is written (*i.e.* its state changed from HRS to LRS or vice versa) by the application of appropriate voltage across the ReRAM cell's electrodes. The voltages V_{SET} and V_{RESET} are characteristics of the device and are determined by the switching oxide, its thickness, electrode material and other such physical properties of the ReRAM cell. *e.g.* a $Ti/SiO_2(5nm)/C$ device has a V_{SET} of 2.4 V and a V_{RESET} of -1.25 V, as reported in [1]. This means that, if the device is in HRS and 2.5 V (or any voltage greater than 2.4 V) is applied, the device will switch to LRS. Similarly, if the device is in LRS, the ReRAM cell is supposed to switch to HRS if any voltage of opposite polarity greater than 1.25 V is applied. However, this is an ideal situation. In practise, V_{SET} and V_{RESET} are not single values, but a range. Fig. 6.1 plots this reality, where the variation in V_{SET} and V_{RESET} is simulated by fitting the Stanford-PKU model to the variability observed in hafnium oxide ReRAMs manufactured at IHP Microelectronics, Germany [2].

This V_{SET} and V_{RESET} variation is a very important phenomenon to be kept in mind while writing to the cell because it affects the design of the WRITE circuitry. The variation has energy implications as well because, as we will see in the coming section, writing is performed by a series of pulses (instead of a single pulse) to minimize variation. To make matters simple, first we start with the basic WRITE circuitry assuming absence of variations. Therefore, the cell can be written by a single pulse. In fact, when ReRAM technology was first explored, the cells were programmed by a single pulse. Later, since programming by a single pulse

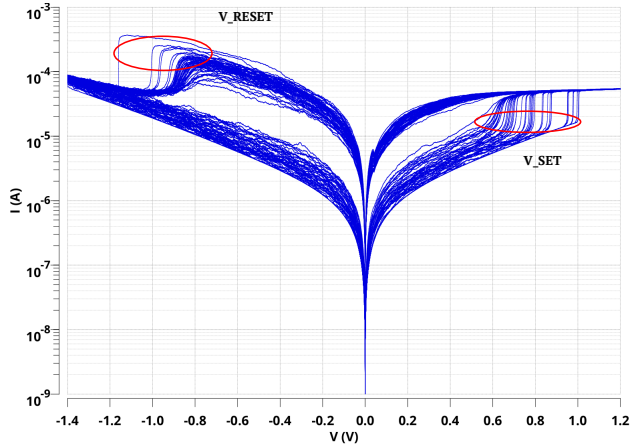


Fig. 6.1 In ReRAM technology, due to cycle-to-cycle variations, the device does not always switch from HRS to LRS and vice versa, at a particular voltage. The threshold voltage above which the device switches is not a single value, but a range. Redrawn from [2].

resulted in a lot of variations, two directions were pursued- incremental programming (series of pulses of progressively increasing amplitude with READ pulses in between) and self-termination programming (the WRITE stimulus is automatically cut-off once the cell is programmed to a target resistance).

6.2 Basic WRITE Circuit

We recall from Chapter 4 that when ReRAM cell is integrated in series with a transistor, it enables better programming since the current through the ReRAM cell is controlled by the transistor. This is especially crucial during SET process when the cell is switched from HRS to LRS. When the cell switches from HRS to LRS, the sudden lowering of resistance can cause a sudden increase in flow of current through the device – a current spike. The transistor avoids this current spike by limiting the current that can flow through the 1T-1R device to a few hundreds of μA, the maximum drain-source current corresponding to the gate voltage of the transistor. Due to the aforementioned reason, ReRAM manufactured in 1T-1R configuration is safer to operate compared to 1S-1R configuration.

resistive switching memory by dynamic self-adaptive write method," in *2013 Symposium on VLSI Technology*, pp. T102–T103, 2013.

[16] M. Alayan, E. Muhr, A. Levisse, M. Bocquet, M. Moreau, E. Nowak, G. Molas, E. Vianello, and J. M. Portal, "Switching event detection and self-termination programming circuit for energy efficient reram memory arrays," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 5, pp. 748–752, 2019.

[17] W.-H. Chen, W.-J. Lin, L.-Y. Lai, S. Li, C.-H. Hsu, H.-T. Lin, H.-Y. Lee, J.-W. Su, Y. Xie, S.-S. Sheu, and M.-F. Chang, "A 16mb dual-mode reram macro with sub-14ns computing-in-memory and memory functions enabled by self-write termination scheme," in *2017 IEEE International Electron Devices Meeting (IEDM)*, pp. 28.2.1–28.2.4, 2017.

[18] Z. Wang, Z. He, R. Yang, S. Fan, J. Lin, F. Liu, Y. Jia, C. Yuan, Q. Tang, and L. Jiang, "Self-terminating write of multi-level cell reram for efficient neuromorphic computing," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1251–1256, 2022.

[19] S. Yu and P.-Y. Chen, "Emerging memory technologies: Recent trends and prospects," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 2, pp. 43–56, 2016.

[20] Z. Chen, H. Li, H.-Y. Chen, B. Chen, R. Liu, P. Huang, F. Zhang, Z. Jiang, H. Ye, B. Gao, L. Liu, X. Liu, J. Kang, H.-S. P. Wong, and S. Yu, "Disturbance characteristics of half-selected cells in a cross-point resistive switching memory array," *Nanotechnology*, vol. 27, p. 215204, apr 2016.

[21] H. Li, H.-Y. Chen, Z. Chen, B. Chen, R. Liu, G. Qiu, P. Huang, F. Zhang, Z. Jiang, B. Gao, L. Liu, X. Liu, S. Yu, H.-S. P. Wong, and J. Kang, "Write disturb analyses on half-selected cells of cross-point rram arrays," in *2014 IEEE International Reliability Physics Symposium*, pp. MY.3.1–MY.3.4, 2014.

Chapter 7
Row/Column Access Circuits and Multilevel Cell

7.1 Row Decoder: Circuit to Select a Row

The function of the row-decoder is to select a row, which needs to be written or read from. Electrically, this is accomplished by driving the row wire to a high voltage. In ReRAM, assuming 1T–1R configuration, selecting a row means that we need to make one particular WL out of n WLs go high. One particular WL being high makes the access transistor to be ON, enabling the flow of current through the ReRAM cell for reading/writing. Conventionally, row decoders accomplish this by a combinatorial logic. While designing a row decoder, it is important to minimize the latency/delay of the decoding process. From the system point of view, for a memory technology to be competitive, the memory access time, *i.e.* the time required to read (which includes the time to select a row) from it must be minimum so that the memory access does not slow down the entire system. Conventionally, row decoders were designed using Boolean gates as a combinational logic. A typical decoder has n inputs and 2^n outputs and for each input combination, only one of the 2^n output signals goes high. Fig. 7.1 illustrates an example of 2:4 decoding using the common types of decoders – static NAND, dynamic NAND and dynamic NOR decoders.

Static NAND decoder simply uses a CMOS NAND gate for each row. Depending on the address (A_1A_0), only one of the four outputs $X_3X_2X_1X_0$ will be LOW and all others will be HIGH. For n-inputs, the decoder will occupy huge area since each row needs an n-input NAND gate. The area will be extremely large as n grows. As depicted in Fig. 7.1, dynamic decoders have a single PMOS transistor in the pull-up network of

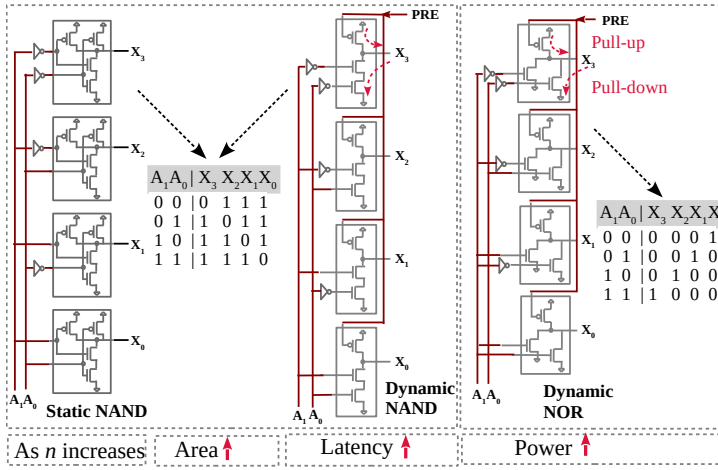


Fig. 7.1 Example of 2:4 decoding using static NAND, dynamic NAND and dynamic NOR decoders. (A_1A_0) represent the input address and $(X_3X_2X_1X_0)$ are the four outputs which are connected to the four rows/ WLs.

each row (thus offering area reduction compared to static decoders). PRE signal is initially LOW and this pre-charges the output node of the decoder to V_{DD}. During the actual decoding, PRE goes high and depending on the inputs (A_1A_0), one of the four outputs is selected. In dynamic NAND decoder, the pull-down network is similar to a CMOS NAND gate (hence the name). Depending on the inputs, one of the four outputs will be pulled down and all other outputs are high. The dynamic NOR decoder has a pull-down network similar to a CMOS NOR gate. It can be verified that for a dynamic NOR decoder, only one of the four outputs remain HIGH and all other outputs are pulled down to ground during decoding. This implies that for a 8:256 dynamic NOR decoder, 255 nodes are pulled down to ground during decoding process. In contrast, for the same decoding task, only a single output node is pulled to ground during NAND decoding. Hence dynamic NOR decoders consume more power when compared to dynamic NOR decoders [1]. However, dynamic NAND decoders are slower than dynamic NOR decoders because the output node is discharged through stacked transistors (as n increases, the output node is discharged through n transistors while it is discharged through a single transistor in dynamic NOR).

Among these types of decoders, dynamic NAND decoder is one feasible row decoder adopted for ReRAM array due to its compact area

32 mb flash memory," in *Proceedings ISSCC '95 - International Solid-State Circuits Conference*, pp. 132–133, 1995.

[11] Y. Yilmaz and P. Mazumder, "A drift-tolerant read/write scheme for multilevel memristor memory," *IEEE Transactions on Nanotechnology*, vol. 16, pp. 1016–1027, Nov 2017.

[12] F. Husain, B. Iqbal, and A. Grover, "A 0.4µa offset, 6ns sensing-time multi-level sense amplifier for resistive non-volatile memories in 65nm lstp technology," in *2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID)*, pp. 76–81, 2021.

[13] J. Reuben and D. Fey, "A time-based sensing scheme for multi-level cell (mlc) resistive ram," in *2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, pp. 1–6, 2019.

[14] X. Zhang, B.-K. An, and T. T.-H. Kim, "A robust time-based multi-level sensing circuit for resistive memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 1, pp. 340–352, 2023.

[15] W. Li, X. Sun, S. Huang, H. Jiang, and S. Yu, "A 40-nm mlc-rram compute-in-memory macro with sparsity control, on-chip write-verify, and temperature-independent adc references," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 9, pp. 2868–2877, 2022.

[16] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao, S. Joshi, H. Wu, H.-S. P. Wong, and G. Cauwenberghs, "A compute-in-memory chip based on resistive random-access memory," *Nature*, vol. 608, no. 7923.

[17] J. M. Rabaey, *Digital integrated circuits: a design perspective*. USA: Prentice-Hall, Inc., 1996.

[18] M. Jagasivamani, C. Walden, D. Singh, L. Kang, S. Li, M. Asnaashari, S. Dubois, B. Jacob, and D. Yeung, "Memory-systems challenges in realizing monolithic computers," in *Proceedings of the International Symposium on Memory Systems*, MEMSYS '18, (New York, NY, USA), p. 98–104, Association for Computing Machinery, 2018.

[19] A. Levisse, p. Royer, B. Giraud, J. Noel, M. Moreau, and J. Portal, "Architecture, design and technology guidelines for crosspoint memories," in *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 55–60, 2017.

[20] P. Narayanan, G. W. Burr, R. S. Shenoy, S. Stephens, K. Virwani, A. Padilla, B. N. Kurdi, and K. Gopalakrishnan, "Exploring the design space for crossbar arrays built with mixed-ionic-electronic-conduction (miec) access devices," *IEEE Journal of the Electron Devices Society*, vol. 3, no. 5, pp. 423–434, 2015.

[21] P. Chen, S. Sheu, K. Cheng, H. Lee, F. T. Chen, P. Chiang, M. Tsai, W. Lin, M. Chang, and Y. Chen, "Fast-write resistive ram (rram) for embedded applications," *IEEE Design & Test of Computers*, vol. 28, pp. 64–71, jan 2011.

[22] S. Yu, "Rram array architecture," in *Resistive Random Access Memory (RRAM): From Devices to Array Architectures*, pp. 35–54, Cham: Springer International Publishing, 2016.

[23] S. Yin, Y. Kim, X. Han, H. Barnaby, S. Yu, Y. Luo, W. He, X. Sun, J.-J. Kim, and J.-s. Seo, "Monolithically integrated rram- and cmos-based in-memory

computing optimizations for efficient deep learning," *IEEE Micro*, vol. 39, no. 6, pp. 54–63, 2019.

[24] J. Reuben and S. Pechmann, "Accelerated addition in resistive ram array using parallel-friendly majority gates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1108–1121, 2021.

Part III
IN–MEMORY COMPUTING

Chapter 8
In-Memory Computing: Applications of Resistive RAM Beyond Memory

8.1 Introduction

The reader might recall the statements made in the introduction of this book about the applications of Resistive RAM technology beyond a pure memory. Specifically, it was discussed how the conventional role of memory is being re-engineered in the field of integrated circuits. Historically, memory was considered a dumb device with no capability to compute or manipulate data which it stores. When you read from a memory, you get what you have written into it before. In other words, the sole purpose of memory (volatile or non-volatile) was storing or keeping the data unaltered in a chip. In DRAM, the charge on the capacitor is kept unaltered and in ReRAM, the resistance remains unaltered. As long as the device can maintain that state unaltered, it satisfied the requirements expected of a memory. Hence, the requirement for a memory was simply a storage unit and the conventional memories satisfied that requirement well, for many decades. Today, what is expected of a memory device in an integrated circuit is not just store data, but also perform some computations. This is due to a phenomenon called 'von Neumann bottleneck' in conventional computer architecture.

8.1.1 Von Neumann Bottleneck (The Memory Wall)

The movement of data between processing and memory units in present day computing systems is their main performance and energy-efficiency bottleneck, often referred to as the 'von Neumann bottleneck' or 'memory

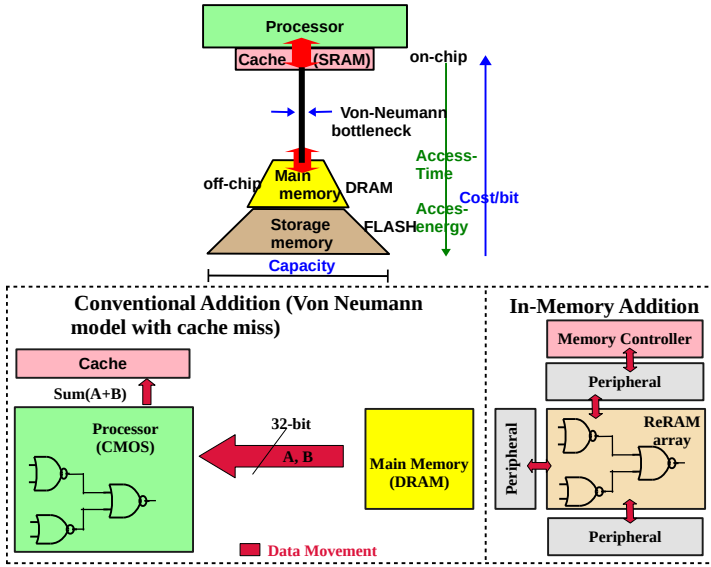


Fig. 8.1 (top) The von Neumann bottleneck. If the data required by the processor is not available in cache, it has to be brought from the main memory through a bandwidth-limited channel (bottom). If two 32-bit numbers have to be added, they have to be fetched from the DRAM and added in a CMOS-based processor in conventional addition. In in-memory addition, the data is moved only around the memory (array, peripheral circuitry and memory controller) and addition is performed in the array.

wall'. The memory wall problem has two facets: the mismatch in the performance (speed) of processor and memory and the energy for data transfer during memory access. The energy to access (transfer) data is growing exponentially along the memory hierarchy (from cache to off–chip DRAM, Fig.8.1-(top)). As a quantitative example, Stanford university researchers [1] point out that the energy for DRAM access is 3556 $\times$ the energy for 16-bit addition in 45 nm CMOS technology. It is widely accepted that 'data movement energy' dominates the 'computation energy' in traditional systems with separate memory and processing units, *i.e.* the computation in itself consumes only a small fraction of the energy [1, 2, 3] . There has been an ongoing effort (since 15-20 years) to combat the memory wall by bringing the processor and memory unit closer to each other. Early researchers used the term processing-in-memory (PIM) to refer to the effort to move the processing closer to where data resides. The term processing-

in-memory broadly referred to processing in the memory using computing units placed in the memory chip. Later researchers used the term near-memory computing or near data processing to refer to the same effort and they exploited 3D stacking of DRAM dies over logic die to compute near memory. However, all these efforts try to minimize the physical distance between memory (where data is stored) and processing units and they do not completely solve the von Neumann bottleneck because the **need for data transfer persists**. The era of big data exacerbates the von Neumann bottleneck. Big data signifies increase in volume (due to increase in number of communication devices), variety (photos, videos, audio recordings, email messages, documents, books, presentations) and velocity of data (data traffic was expected to grow by a factor of 1455 from 2018 to 2020) [4]. This increase in data poses challenges to the conventional von Neumann architecture where memory and processor are separated.

The memory wall faced by computer architects necessitated a paradigm shift in the way data is processed. At the moment, there is an increasing trend to move computing to the location of data, a memory-centric computing [5, 6, 7]. The term 'in-memory computing' (also called 'processing-in-memory','compute-in-memory', 'logic-in-memory') is used to refer to any effort to process data at the residence of data (*i.e.* in the memory array) without moving them to a separate CMOS-based processing unit. If one can compute at the residence of data, in principle, the memory wall problem could be solved since the demarcation (between processing unit and memory) and the accompanying costly data transfer can be eliminated. The data transfer between processor and memory is costly in terms of energy and latency. Furthermore, when we move computing from processor to the memory, the substrate on which computing is performed is usually the memory array. The regular structure of the memory array (rows and columns arrangement of memory cells) is advantageous for parallel-processing and may even aid in reducing the latency of computation in certain cases, *i.e.* numerous operations can be performed simultaneously in the columns of a memory array which is not possible in typical CMOS substrate constructed from heterogeneous gates. In-memory computing is being explored for a wide variety of applications like scientific computing (linear and partial differential equations), machine learning, signal/image processing and pattern recognition. In all these applications, arithmetic (addition/subtraction) serves as the foundation. Detailed survey of in-memory computing using emerging NVMs are presented in [5, 7, 6, 8, 9]. In this chapter, we will limit our discussions to 'In-memory Arithmetic' and 'Matrix Vector Multiplication'.

8.2 In-memory Arithmetic

The author uses the term "arithmetic" as it is used in computer arithmetic. However, here the arithmetic is performed completely in the memory array. Addition, being the basic component of all computer arithmetic, this section will survey adders implemented in ReRAM array. Boolean logic has been the predominant way of designing adders in CMOS technology. Adders were synthesized in terms of logic gates, optimized using logic synthesis tools and mapped to 'x' nm node during technology mapping. Each gate is implemented as a pull-up (PMOS) and pull-down (NMOS) network of MOS transistors. Hence, the atomic computing unit in CMOS adders is the transistor (the latency and switching energy of transistors dictated the speed and power consumption of the adder). When researchers wanted to move arithmetic to memory, they followed a different path. They pursued a method to implement a Boolean logic gate using memory cell(s) of the array while minimally modifying the array and/or its peripherals. Adders could then be implemented as a sequence of Boolean logic operations in this array. This implies that the atomic computing unit of in-memory adders is the logic primitive or the Boolean gate (there may be exceptions). Hence, logic primitive plays a significant role in the performance of in-memory adders. In addition to logic primitive, the way in which carry is propagated (ripple carry, carry look-ahead, parallel-prefix *etc.*) also plays a crucial role in the performance of in-memory adders. The latency of different adders heavily depends on the method of carry-propagation.

8.2.1 Boolean Logic Gates in ReRAM array

As stated, adders are implemented in memory array by implementing Boolean logic primitive in the memory array. To implement complex functions like adders, the Boolean logic primitive chosen must be 'functionally complete'. A logic primitive or a set of logic primitives is said to be functionally complete if any Boolean function can be implemented in terms of this logic primitive (s). For example, NOR is 'functionally complete' since any Boolean function can be expressed in terms of NOR logic primitive*. Therefore, if a NOR gate can be implemented in a ReRAM array, any Boolean function can be implemented in the ReRAM array. We call this

* Recall from Digital Logic course that NAND and NOR gates were called universal gates for this reason.

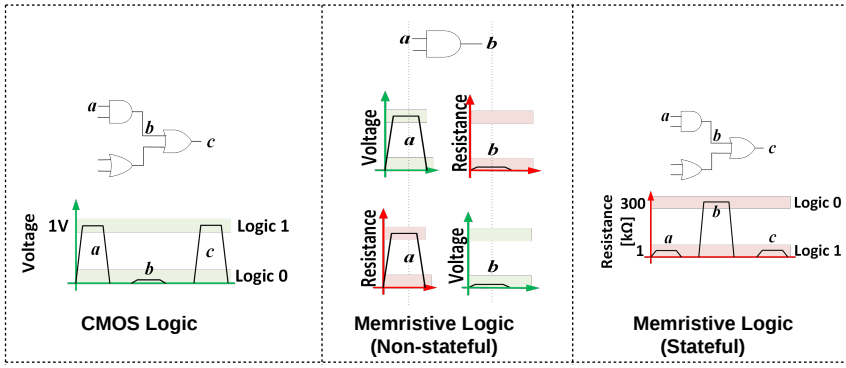


Fig. 8.2 In CMOS logic, inputs, outputs and intermediate values are represented as voltages; in memristive logic, inputs, outputs and intermediate values are represented as either voltages or resistances. A memristive logic is said to be stateful if resistance is the only state variable for representing inputs, outputs and intermediate results of computation.

type of logic 'memristive logic' since a memristive device (Resistive RAM) was used to construct it. Other than NOR, NAND, IMPLY+FALSE [10] and Majority+NOT [11] are also functionally complete. Before we discuss the implementation of some memristive logic gates, we must clarify an important difference between these memristive logic gates and CMOS logic gates. In CMOS logic, there is only a single logic state variable, *i.e.*, voltage. The input data is represented as voltage and is processed as voltage throughout the computation (including in all of the intermediate stages), and is finally also represented as voltage at the output, as illustrated in Fig.8.2. Furthermore, the state variable is regenerated throughout the computation by the CMOS gates. This seamless flow is disrupted in memristive logic gate because the internal state of memory cells governs their resistance, introducing resistance, in addition to voltage, as a logic state variable for computation [12]. When Boolean logic is implemented by manipulating the state (resistance) of the memristor, it is called stateful logic [13]. In a stateful logic family, both the input and the output of the computation is resistance. In certain logic families, the input is a voltage, but the output is a resistance [14] and in certain other families, the input is the resistance and the output of computation is a voltage [15]. These are classified as non-stateful logic families (Fig. 8.2). Due to space constraints, we discuss only one example for stateful logic and non-stateful logic here.

bit of W happens in an analog manner. Fig.8.16 illustrates how the bit value is applied to the rows of the memory array. Since we do not want to disturb the state of the cell while performing MVM, a bit value of '1' is usually translated as a low voltage of 0.2 V or 0.3 V (usually the READ voltage of the ReRAM technology) and a bit value of '0' is translated as just 0 V. Hence (0 1 1 0) at the row is applied as (0 0.2 0.2 0) Volts. If the cell is '0', it is 200 kΩ and we have a current of 1 μA across the cell when the WL is biased at 0.2 V. If the cell is '1', it is 10 kΩ and we have a current of 20 μA across the cell (assuming the BL is held at virtual ground as before). In this manner, multiplication of bit '1' $\times$ '1' results in 20 μA (representing '1') and multiplication of bit '1' $\times$ '0' results in 1 μA (representing '0'). If the size of the array is reasonable (128$\times$128), these currents get added and a column with more '1'$\times$'1' will result in a larger current. The added BL current is fed to the I-to-V converter and converted to a digital output using the ADC. For the 4$\times$3 array depicted in Fig.8.16, a 3-bit ADC is enough since the maximum value of a column is 4 (1+1+1+1). Notice that we need a simpler ADC (3-bit instead of the 6-bit ADC used in Fig.8.13) since each memory cell represents a single bit of weight W. Furthermore, the DAC used in Fig.8.16 is also less complicated since it needs to apply only a single voltage (0.2 V). In this manner, bit slicing reduces the complexity of the peripheral ADCs and DACs.

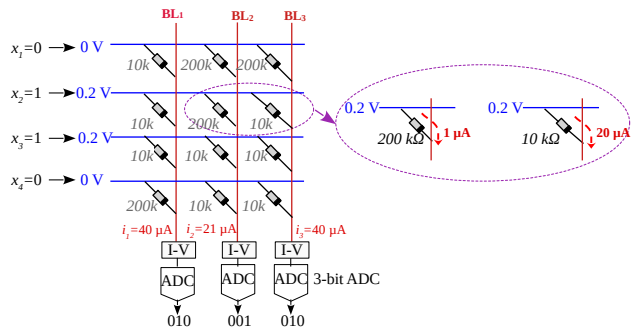


Fig. 8.16 During MVM, in the memory array, the bits are represented as analog values, X as a voltage and W as a resistance.

Since the ReRAM cells are programmed to only two states (HRS and LRS), it can generally be accomplished in a precise manner compared to programming the cells to 2^n states (as needed in pure analog MVM discussed earlier). Therefore the error incurred during MVM due to

imprecise programming of the resistive states is overcome. Consequently bit slicing can overcome the accuracy limitation of pure analog NVM to a certain extent. The requirement of binary programming of the states also make the peripheral circuitry (WRITE circuit) a lot less complicated. However, extra shift-and-add circuits are required and they could increase the energy consumption and area of the peripheral circuitry. Note that a shift-and-add circuit is needed for each column of the weight matrix W (only a single column is depicted in Fig.8.15). Finally, latency of MVM using bit slicing is increased when compared to pure analog MVM (Fig.8.13) since the input vector X is sliced to n bits and this requires n multiply operations in the memory array followed by shift-and-add operations.

8.4 Conclusion and Future Outlook

One must be cautioned that all processing should not be moved to memory since it is not always beneficial. There are applications where in-memory computing performs worse when compared to pure-CMOS implementation. For example, performing addition and multiplication in memory in a digital fashion using Boolean logic operations may incur hundreds and few thousands memory cycles as the bit-width increases from 32-bit to 64-bit [22, 62]. In this case, in-memory addition/multiplication makes sense only when the combined cost(energy and delay) to move the inputs from the memory to the processor and compute in a CMOS-based processor is more than the cost of in-memory addition. What is really worth pursuing is a hybrid approach- perform parts of the processing in CMOS (when it is 'better' to compute there) and remaining parts in the memory (when it is 'better' to compute there).

To give an example from the automobile world, computing in future should take an approach similar to Toyota's hybrid technology for driving. With Prius and its successors, Toyota pioneered using both an internal combustion engine and an electric motor. Since the combustion engines have an optimum temperature to which they must be heated (before they can become efficient), the car begins to start moving with the electric motor in a pure electric mode. Then, as the speed increases it switches to combustion engine. If the car is slowed down due to a traffic jam, its switches to electric mode automatically till the queue is cleared. In this manner, the hybrid technology uses the best of both worlds to achieve very good fuel efficiency. The key to achieving fuel efficiency is to exploit the

strength of the different technologies at the right time. To move the vehicle, a high torque is needed which an electric motor provides very easily. Using pure electric drive at low speeds is also beneficial for reducing pollution in cities and city limits. On the other hand, range anxiety (number of kms that can be driven before battery loses charge) and the high electric power consumption at high speeds make pure electric drive disadvantageous at high speeds. For high speeds and long distance driving, the combustion engine is advantageous over pure electric drive[¶].

The crucial question to be answered is: how do we know when it is 'better' to compute in CMOS technology? and when it is 'better' to compute in the memory array? This question is not easily answered but depends on the application and also on the specific type of NVM technology considered. It must be noted that different NVMs vary widely in their characteristics like endurance, retention, switching time *etc.* Generally, in applications where there is heavy data movement between processor and memory, computing in memory is proven to be energy efficient [6]. If the computing in ReRAM array is performed in an analog fashion like the MVM considered in this chapter, the accuracy is degraded due to cycle-to-cycle and device-to-device variability of ReRAM cells. Therefore, MVM in ReRAM array in an analog manner may not be suitable in applications like scientific computing where a high degree of accuracy is required. On the other hand, there are applications like image processing where an approximate solution is considered acceptable. This is because, a higher accuracy in certain image-processing tasks, even if achievable, is not recognisable to the human eye. In conclusion, computing of future should take a hybrid approach and exploit the advantages of CMOS technology and NVM technology judiciously. The future of computing with the advent of ReRAM and other similar NVM technology will be exciting.

References

[1] A. Pedram, S. Richardson, M. Horowitz, S. Galal, and S. Kvatinsky, "Dark memory and accelerator-rich system optimization in the dark silicon era,"

[¶] In reality, Toyota's hybrid technology is more complex since it uses the synergy between electric and combustion technology and there are situations when both work **together** to optimize fuel. However, the purpose of using Toyota's hybrid car analogy here was to drive home the point that computing needs to be performed in a hybrid manner depending on whether it is energy-efficient to be performed in CMOS technology or in the memory array.

IEEE Design Test, vol. 34, pp. 39–50, April 2017.
[2] O. Mutlu, S. Ghose, J. Gómez-Luna, and R. Ausavarungnirun, "Processing data where it makes sense: Enabling in-memory computation," *Microprocessors and Microsystems*, vol. 67, pp. 28–41, 2019.
[3] W. J. Dally, Y. Turakhia, and S. Han, "Domain-specific hardware accelerators," *Commun. ACM*, vol. 63, p. 48–57, jun 2020.
[4] G. Molas and E. Nowak, "Advances in emerging memory technologies: From data storage to artificial intelligence," *Applied Sciences*, vol. 11, no. 23, 2021.
[5] D. Ielmini and H.-S. P. Wong, "In-memory computing with resistive switching devices," *Nature Electronics*, vol. 1, pp. 333 – 343, 2018.
[6] A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, 2020.
[7] H. A. D. Nguyen, J. Yu, M. A. Lebdeh, M. Taouil, S. Hamdioui, and F. Catthoor, "A classification of memory-centric computing," *J. Emerg. Technol. Comput. Syst.*, vol. 16, Jan. 2020.
[8] N. Xu, T. Park, K. J. Yoon, and C. S. Hwang, "In-memory stateful logic computing using memristors: Gate, calculation, and application," *physica status solidi (RRL) – Rapid Research Letters*, vol. 15, no. 9, p. 2100208, 2021.
[9] Y. S. Kim, M. W. Son, and K. M. Kim, "Memristive stateful logic for edge boolean computers," *Advanced Intelligent Systems*, vol. 3, no. 7, p. 2000278, 2021.
[10] S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Memristor-based material implication (imply) logic: Design principles and methodologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, pp. 2054–2066, Oct 2014.
[11] L. Amarú, P. E. Gaillardon, and G. D. Micheli, "Majority-inverter graph: A new paradigm for logic optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, pp. 806–819, May 2016.
[12] J. Reuben, R. Ben-Hur, N. Wald, N. Talati, A. Ali, P.-E. Gaillardon, and S. Kvatinsky, "Memristive logic: A framework for evaluation and comparison," in *Power And Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 1–8, September 2017.
[13] E. Lehtonen, J. H. Poikonen, and M. Laiho, "Memristive stateful logic," in *Memristor Networks* (A. Adamatzky and L. Chua, eds.), pp. 603–623, Cham: Springer International Publishing, 2014.
[14] S. Shirinzadeh, M. Soeken, P. Gaillardon, and R. Drechsler, "Fast logic synthesis for rram-based in-memory computing using majority-inverter graphs," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 948–953, 2016.
[15] S. Li, C. Xu, Q. Zou, J. Zhao, Y. Lu, and Y. Xie, "Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories," in *Proceedings of the 53rd Annual Design Automation Conference*, DAC '16, (New York, NY, USA), pp. 173:1–173:6, ACM, 2016.
[16] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "Magic—memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.

[17] B. Hoffer, V. Rana, S. Menzel, R. Waser, and S. Kvatinsky, "Experimental demonstration of memristor-aided logic (magic) using valence change memory (vcm)," *IEEE Transactions on Electron Devices*, vol. 67, no. 8, pp. 3115–3122, 2020.

[18] E. Testa, M. Soeken, L. G. Amaru, and G. De Micheli, "Logic synthesis for established and emerging computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 165–184, 2019.

[19] G. Jaberipur, B. Parhami, and D. Abedi, "Adapting computer arithmetic structures to sustainable supercomputing in low-power, majority-logic nanotechnologies," *IEEE Transactions on Sustainable Computing*, vol. 3, pp. 262–273, Oct 2018.

[20] V. Pudi, K. Sridharan, and F. Lombardi, "Majority logic formulations for parallel adder designs at reduced delay and circuit complexity," *IEEE Transactions on Computers*, vol. 66, pp. 1824–1830, Oct 2017.

[21] J. Reuben and S. Pechmann, "A parallel-friendly majority gate to accelerate in-memory computation," in *2020 IEEE 31st International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 93–100, 2020.

[22] J. Reuben and S. Pechmann, "Accelerated addition in resistive ram array using parallel-friendly majority gates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1108–1121, 2021.

[23] J. Reuben, "Binary addition in resistance switching memory array by sensing majority," *Micromachines*, vol. 11, no. 5, 2020.

[24] S. Pechmann, T. Mai, M. Völkel, M. K. Mahadevaiah, E. Perez, E. Perez-Bosch Quesada, M. Reichenbach, C. Wenger, and A. Hagelauer, "A versatile, voltage-pulse based read and programming circuit for multi-level rram cells," *Electronics*, vol. 10, no. 5, 2021.

[25] H. Padberg, A. Regev, G. Piccolboni, A. Bricalli, G. Molas, J. F. Nodin, and S. Kvatinsky, "Experimental demonstration of non-stateful in-memory logic with 1t1r oxram valence change mechanism memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2023.

[26] L. Brackmann, T. Ziegler, A. Jafari, D. J. Wouters, M. B. Tahoori, and S. Menzel, "Improved arithmetic performance by combining stateful and non-stateful logic in resistive random access memory 1t–1r crossbars," *Advanced Intelligent Systems*, vol. 6, no. 3, p. 2300579, 2024.

[27] R. Ben Hur, N. Wald, N. Talati, and S. Kvatinsky, "Simple magic: Synthesis and in-memory mapping of logic execution for memristor-aided logic," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 225–232, 2017.

[28] R. Ben-Hur, R. Ronen, A. Haj-Ali, D. Bhattacharjee, A. Eliahu, N. Peled, and S. Kvatinsky, "Simpler magic: Synthesis and mapping of in-memory logic executed in a single row to improve throughput," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2434–2447, 2020.

[29] Rumi Zhang, K. Walus, Wei Wang, and G. A. Jullien, "A method of majority logic reduction for quantum cellular automata," *IEEE Transactions on Nanotechnology*, vol. 3, pp. 443–450, Dec 2004.

[30] S. Sheu, K. Cheng, M. Chang, P. Chiang, W. Lin, H. Lee, P. Chen, Y. Chen, T. Wu, F. T. Chen, K. Su, M. Kao, and M. Tsai, "Fast-write resistive ram (rram) for embedded applications," *IEEE Design Test of Computers*, vol. 28, pp. 64–71, Jan 2011.

[31] P. Huang, J. Kang, Y. Zhao, S. Chen, R. Han, Z. Zhou, Z. Chen, W. Ma, M. Li, L. Liu, and X. Liu, "Reconfigurable nonvolatile logic operations in resistance switching crossbar array for large-scale circuits," *Advanced Materials*, vol. 28, no. 44, pp. 9758–9764, 2016.

[32] A. Karimi and A. Rezai, "Novel design for a memristor-based full adder using a new imply logic approach," *J Comput Electron*, vol. 17, p. 11303–1314, 2018.

[33] L. Cheng *et al.*, "Functional demonstration of a memristive arithmetic logic unit (memalu) for in-memory computing," *Advanced Functional Materials*, vol. 29, no. 49, p. 1905660, 2019.

[34] S. Ganjeheizadeh Rohani, N. Taherinejad, and D. Radakovits, "A semiparallel full-adder in imply logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 1, pp. 297–301, 2020.

[35] N. Talati, S. Gupta, P. Mane, and S. Kvatinsky, "Logic design within memristive memories using memristor-aided logic (magic)," *IEEE Transactions on Nanotechnology*, vol. 15, no. 4, pp. 635–650, 2016.

[36] Y. S. Kim, M. W. Son, H. Song, J. Park, J. An, J. B. Jeon, G. Y. Kim, S. Son, and K. M. Kim, "Stateful in-memory logic system and its practical implementation in a taox-based bipolar-type memristive crossbar array," *Advanced Intelligent Systems*, vol. 2, no. 3, p. 1900156, 2020.

[37] K. Alhaj Ali *et al.*, "Memristive computational memory using memristor overwrite logic (mol)," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2370–2382, 2020.

[38] A. Siemon, R. Drabinski, M. J. Schultis, X. Hu, E. Linn, A. Heittmann, R. Waser, D. Querlioz, S. Menzel, and J. S. Friedman, "Stateful three-input logic with memristive switches," *Scientific Reports*, vol. 9, no. 1, p. 14618, 2019.

[39] A. Siemon, S. Menzel, R. Waser, and E. Linn, "A complementary resistive switch-based crossbar array adder," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 5, no. 1, pp. 64–74, 2015.

[40] N. TaheriNejad, "Sixor: Single-cycle in-memristor xor," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–11, 2021.

[41] F. Pinto and I. Vourkas, "Robust circuit and system design for general-purpose computational resistive memories," *Electronics*, vol. 10, no. 9, 2021.

[42] Z. Wang *et al.*, "Efficient implementation of boolean and full-adder functions with 1t1r rrams for beyond von neumann in-memory computing," *IEEE Transactions on Electron Devices*, vol. 65, pp. 4659–4666, Oct 2018.

[43] A. Siemon *et al.*, "Sklansky tree adder realization in 1s1r resistive switching memory architecture," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2269–2285, 2019.

[44] J. Reuben, "Rediscovering majority logic in the post-cmos era: A perspective from in-memory computing," *Journal of Low Power Electronics and Applications*, vol. 10, no. 3, 2020.

[45] P. Wang, M. Y. Niamat, S. R. Vemuru, M. Alam, and T. Killian, "Synthesis of majority/minority logic networks," *IEEE Transactions on Nanotechnology*, vol. 14, no. 3, pp. 473–483, 2015.

[46] C.-C. Chung, Y.-C. Chen, C.-Y. Wang, and C.-C. Wu, "Majority logic circuits optimisation by node merging," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 714–719, 2017.

[47] H. Riener, E. Testa, L. Amaru, M. Soeken, and G. D. Micheli, "Size optimization of migs with an application to qca and stmg technologies," in *2018 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 1–6, 2018.

[48] R. Devadoss, K. Paul, and M. Balakrishnan, "Majority logic: Prime implicants and n-input majority term equivalence," in *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*, pp. 464–469, 2019.

[49] A. Neutzling, F. S. Marranghello, J. M. Matos, A. Reis, and R. P. Ribas, "maj-n logic synthesis for emerging technology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 3, pp. 747–751, 2020.

[50] M. Kaneko, "A novel framework for procedural construction of parallel prefix adders," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2019.

[51] C. L. Ayala, N. Takeuchi, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa, "Majority-logic-optimized parallel prefix carry look-ahead adder families using adiabatic quantum-flux-parametron logic," *IEEE Transactions on Applied Superconductivity*, vol. 27, pp. 1–7, June 2017.

[52] J. Reuben, "Design of in-memory parallel-prefix adders," *Journal of Low Power Electronics and Applications*, vol. 11, no. 4, 2021.

[53] A. Levisse, B. Giraud, J. . Noel, M. Moreau, and J. . Portal, "Rram crossbar arrays for storage class memory applications: Throughput and density considerations," in *2018 Conference on Design of Circuits and Integrated Systems (DCIS)*, pp. 1–6, Nov 2018.

[54] A. Amirsoleimani, F. Alibart, V. Yon, J. Xu, M. R. Pazhouhandeh, S. Ecoffey, Y. Beilliard, R. Genov, and D. Drouin, "In-memory vector-matrix multiplication in monolithic complementary metal–oxide–semiconductor-memristor integrated circuits: Design choices, challenges, and perspectives," *Advanced Intelligent Systems*, vol. 2, no. 11, p. 2000115, 2020.

[55] R. Nägele, J. Finkbeiner, V. Stadtlander, M. Grözing, and M. Berroth, "Analog multiply-accumulate cell with multi-bit resolution for all-analog ai inference accelerators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 9, pp. 3509–3521, 2023.

[56] Z. Sun and D. Ielmini, "Invited tutorial: Analog matrix computing with crosspoint resistive memory arrays," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 7, pp. 3024–3029, 2022.

[57] M. Hu, C. E. Graves, C. Li, Y. Li, N. Ge, E. Montgomery, N. Davila, H. Jiang, R. S. Williams, J. J. Yang, Q. Xia, and J. P. Strachan, "Memristor-based analog computation and neural network classification with a dot product engine," *Advanced Materials*, vol. 30, no. 9, p. 1705914, 2018.

[58] S. Agarwal, T.-T. Quach, O. Parekh, A. H. Hsia, E. P. DeBenedictis, C. D. James, M. J. Marinella, and J. B. Aimone, "Energy scaling advantages of resistive memory crossbar based computation and its application to sparse coding," *Frontiers in Neuroscience*, vol. 9, 2016.

[59] K. Roy, I. Chakraborty, M. Ali, A. Ankit, and A. Agrawal, "In-memory computing in emerging memory technologies for machine learning: An overview," in *Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference*, DAC '20, IEEE Press, 2020.

[60] Y. Luo, S. Wang, P. Zuo, Z. Sun, and R. Huang, "Modeling and mitigating the interconnect resistance issue in analog rram matrix computing circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 11, pp. 4367–4380, 2022.

[61] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars," *SIGARCH Comput. Archit. News*, vol. 44, p. 14–26, June 2016.

[62] V. Lakshmi, J. Reuben, and V. Pudi, "A novel in-memory wallace tree multiplier architecture using majority logic," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 3, pp. 1148–1158, 2022.

Also by John Reuben

Non-Volatile Memories: Device and Peripheral Circuit Design
Covering Flash, Resistive-switching, and Ferroelectric Memories

(Expected Publication: June 2026)

This comprehensive book is written for beginners to introduce them to non-volatile memories, which will form an integral part of future computing systems. AI hardware will be predominantly memory-centric with computation being performed in memory or near memory arrays. After briefly introducing SRAM and DRAM, this book will introduce the reader to floating gate MOSFET memory (flash), the current standard for non-volatile memories. Then, this book will discuss Resistive-switching memories (ReRAM, Phase Change Memory, STT/SOT-MRAM) and ferroelectric memories (FeRAM, FTJ, FeFET). For each of these memories, the device stucture, array configuration and READ and WRITE process are described in detail with suitable circuits. The book concludes with a chapter on in-memory computing so that readers can appreciate how these memories are used for Vector Matrix Multiplication, the fundamental computation in Deep Neural Networks.

John Reuben is a researcher at Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany. He is a recipient of the Viterbi Post-doctoral fellowship, Technion, Israel, and Emerging Talents Initiative, FAU, Germany. His research focus is emerging Non-Volatile Memories and their applications to in-memory computing. He has published extensively in this field, notably in IEEE Transactions and leading IEEE conferences worldwide.

This book is written as an introductory textbook on Resistive Random Access Memory (ReRAM). ReRAM is a prominent emerging memory among other competing Non-Volatile Memories (NVM) seeking to replace flash memory. This book is based on the author's peer-reviewed research conducted at the Chair of Computer Architecture, FAU, Germany. Referring to his research and the most relevant research from the literature, the author presents the development in this field concisely. The purpose is to clarify basic concepts and introduce the reader to ReRAM with an emphasis on circuit design. Hence, this book is written for university students considering a career in the semiconductor industry. Since the author's research was conducted in collaboration with a silicon foundry, hardware engineers will find this book practical and industry-relevant. Researchers in the field of In-Memory Computing will also benefit from this book since the NVM array is the basic substrate for such computing paradigms.

This three-part book condenses the research and development of the last decade into eight chapters. In Part I, a good foundation is laid to understand the individual device structure, its electrical characteristics, and modeling methodology. The different array configurations in which these memory devices are fabricated are also discussed. In Part II, the peripheral circuits—the CMOS circuits around the ReRAM array are discussed. They include sense amplifiers, programming circuits, and row/column access circuits. Recent developments such as the possibility to perform certain computing tasks in a ReRAM array are discussed in Part III.

ISBN 978-3-00-077848-3